Large Margin Nearest Neighbor Embedding for Knowledge Representation

Miao Fan[†,‡,*], Qiang Zhou[†], Thomas Fang Zheng[†] and Ralph Grishman[‡]
[†] CSLT, Division of Technical Innovation and Development,
Tsinghua National Laboratory for Information Science and Technology,
Tsinghua University, Beijing, 100084, China.
[‡]Proteus Group, New York University, NY, 10003, U.S.A.
[*]fanmiao.cslt.thu@gmail.com

***Abstract*—Traditional way of storing facts in triplets (*head_entity, relation, tail_entity*), abbreviated as (*h, r, t*), makes the knowledge intuitively displayed and easily acquired by mankind, but hardly computed or even reasoned by AI machines. Inspired by the success in applying *Distributed Representations* to AI-related fields, recent studies expect to represent each entity and relation with a unique low-dimensional embedding, which is different from the symbolic and atomic framework of displaying knowledge in triplets. In this way, the knowledge computing and reasoning can be essentially facilitated by means of a simple *vector calculation*, i.e. $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. We thus contribute an effective model to learn better embeddings satisfying the formula by pulling the positive tail entities $\mathbf{t}^+$ to get together and close to h + r (*Nearest Neighbor*), and simultaneously pushing the negatives $\mathbf{t}^-$ away from the positives $\mathbf{t}^+$ via keeping a *Large Margin*. We also design a corresponding learning algorithm to efficiently find the optimal solution based on *Stochastic Gradient Descent* in iterative fashion. Quantitative experiments illustrate that our approach can achieve the state-of-the-art performance, compared with several latest methods on some benchmark datasets for two classical applications, i.e. *Link prediction* and *Triplet classification*. Moreover, we analyze the parameter complexities among all the evaluated models, and analytical results indicate that our model needs fewer computational resources on outperforming the other methods.**

I. Introduction

Owing to the long-term efforts on distilling the explosive information on the Web, several large scale Knowledge Bases (KBs), such as WordNet[1] [1], OpenCyc[2] [2], YAGO[3] [3] and Freebase[4] [4], have already been built. Despite the different domains these KBs serve for, nearly all of them concentrate on enriching *entities* and *relations*. To facilitate storing, displaying and even retrieving knowledge, we use the highly structured form, i.e. (*head_entity, relation, tail_entity*), for knowledge representation. Each triplet is called a *fact*. So far, some general domain KBs, such as Freebase[5] and YAGO[6], have contained millions of entities and billions of facts, and they should have led a huge leap forward for some canonical AI-related tasks, e.g. *Question Answering Systems*. However, it is realised that this symbolic and atomic framework of representing knowledge makes it difficult to be utilized by most of AI-machines, especially of those which are dominated by statistical approaches.

Recently, many AI-related fields, such as *Image Classification* [5], *Natural Language Understanding* [6] and *Speech Recognition* [7], have made significant progress by means of learning *Distributed Representations*. Taking an example of distributed representations of words applied to *Statistical Language Modeling* [8], it has achieved considerable success by grouping similar words which are close to each other in low-dimensional vector spaces. Moreover, Mikolov et al. [9], [10] discovered somewhat surprising patterns that the learnt word embeddings, to some extent, implicitly capture syntactic and semantic regularities in language. For example, the result of vector calculation $\mathbf{v}_{Madrid} - \mathbf{v}_{Spain} + \mathbf{v}_{France}$ is closer to $\mathbf{v}_{Paris}$ than to any other words [11].

Inspired by the idea of word vector calculation, we look forward to making knowledge computable. If we ideally consider the example mentioned above, the most probable reason $\mathbf{v}_{Madrid} - \mathbf{v}_{Spain} + \mathbf{v}_{France} \approx \mathbf{v}_{Paris}$, is that *capital_city_of* is the relation between *Madrid* and *Spain* , and so is *Paris* and *France*. In other words, we can conclude that:

- There are two facts/triplets, i.e. (*Madrid*, *capital_city_of*, *Spain*) and (*Paris*, *capital_city_of*, *France*).
- We also derive the approximate equation that $\mathbf{v}_{Spain} - \mathbf{v}_{Madrid} \approx \mathbf{v}_{France} - \mathbf{v}_{Paris}$ from $\mathbf{v}_{Madrid} - \mathbf{v}_{Spain} + \mathbf{v}_{France} \approx \mathbf{v}_{Paris}$.

The shared relation *capital_city_of* may help establish the approximate equation due to certain implicit connection. If we assume that the relation *capital_city_of* can also be explicitly represented by a vector, i.e. $\mathbf{v}_{capital\_city\_of}$, the connection will be $\mathbf{v}_{Spain} - \mathbf{v}_{Madrid} \approx \mathbf{v}_{France} - \mathbf{v}_{Paris} \approx \mathbf{v}_{capital\_city\_of}$. Therefore, the fact (*Madrid*, *capital_city_of*, *Spain*) can be modeled in another way, i.e. $\mathbf{v}_{Madrid} + \mathbf{v}_{capital\_city\_of} \approx \mathbf{v}_{Spain}$.

Generally speaking, this paper explores a better approach on knowledge representation by means of learning a unique low-dimensional embedding for each entity and relation in KBs, so that each fact (*head_entity, relation, tail_entity*) can

[1] http://www.princeton.edu/wordnet

[2] http://www.cyc.com/platform/opencyc

[3] www.mpi-inf.mpg.de/yago-naga/yago

[4] http://www.freebase.com

[5] According to the statistics released by the official site, Freebase contains 43 million entities and 1.9 billion triplets for now.

[6] As of 2012, YAGO2s has knowledge of more than 10 million entities and contains more than 120 million facts about these entities.

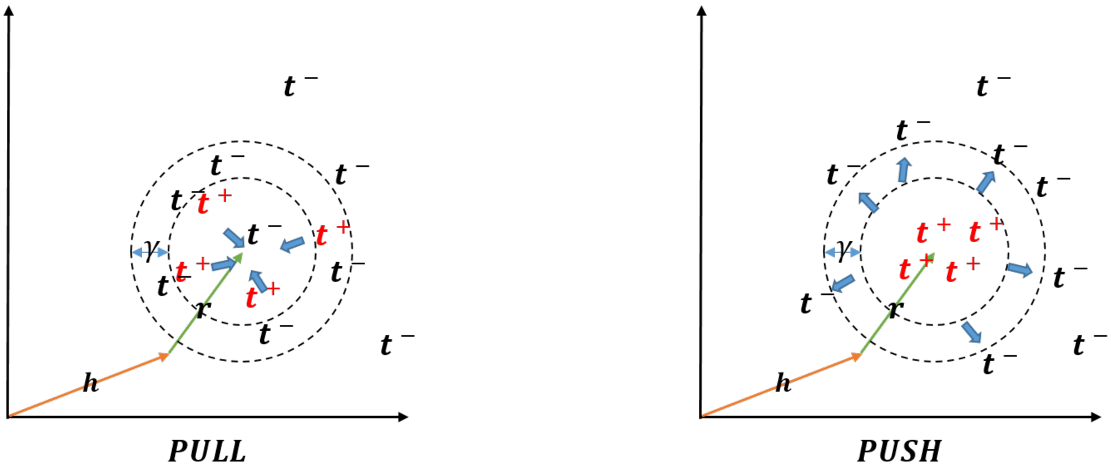

Fig. 1. **LMNNE** has two objects optimized simultaneously: *PULL* the positive tail entities ($\mathbf{t}^+$) close to $\mathbf{h}+\mathbf{r}$ and *PUSH* the negative tail entities ($\mathbf{t}^-$) out of the margin γ. In this case, all the entities and relations are embedded into the 2D vector space and we use L_2 *norm* to measure the distance.

be represented by a simple vector calculation $\mathbf{h}+\mathbf{r} \approx \mathbf{t}$. To achieve this goal, we contribute a generic model named *Large Margin Nearest Neighbor Embedding* (**LMNNE**). As intuitively shown by Figure 1, **LMNNE** follows the principle of *pulling* the positive tail entities $\mathbf{t}^+$ close to $\mathbf{h}+\mathbf{r}$, and simultaneously *pushing* the negative tail entities ($\mathbf{t}^-$) a large margin γ away from the positives $\mathbf{t}^+$. The details about model formulation are described in Section 3. Section 4 presents the algorithm that solves **LMNNE** efficiently based on *Stochastic Gradient Descent* (**SGD**) in iterative fashion. To prove both effectiveness and efficiency of the proposed model, we conduct quantitative experiments in Section 5, i.e. evaluating the performance of *Link prediction* and *Triplet classification* on several benchmark datasets. We also perform qualitative analysis via comparing model complexity among all related approaches in Section 6. Results demonstrate that **LMNNE** can achieve the state-of-the-art performance and demand for fewer computational resources compared with many prior arts.

II. Related Work

All the related studies work on studying better way of representing a fact/triplet. Usually, they design various scoring functions $f_r(h,t)$ to measure the plausibility of a triplet (h,r,t). The lower dissimilarity of the scoring function $f_r(h,t)$ is, the higher compatibility of the triplet (h,r,t) will be.

Unstructured [12] is a naive model which just exploits the occurrence information of the head and the tail entities without considering the relation between them. It defines a scoring function $||\mathbf{h}-\mathbf{t}||$, and this model obviously can not discriminate entity-pairs with different relations. Therefore, **Unstructured** is commonly regarded as the baseline approach.

Distance Model (SE) [13] uses a pair of matrices (W_{rh}, W_{rt}), to characterize a relation r. The dissimilarity of a triplet (h,r,t) is calculate by the L_1 *norm* of $||W_{rh}\mathbf{h}-W_{rt}\mathbf{t}||$. As pointed out by Socher et al. [14], the separating matrices W_{rh} and W_{rt} weaken the capability of capturing correlations between entities and corresponding relations, even though the model takes the relations into consideration.

Single Layer Model proposed by Socher et al. [14] aims at alleviating the shortcomings of **Distance Model** by means of the nonlinearity of a single layer neural network $g(W_{rh}\mathbf{h}+W_{rt}\mathbf{t}+\mathbf{b}_r)$, in which $g = tanh$. Then the linear output layer gives the scoring function: $\mathbf{u}_r^T g(W_{rh}\mathbf{h}+W_{rt}\mathbf{t}+\mathbf{b}_r)$.

Bilinear Model [15], [16] is another model that tries to fix the issue of weak interaction between the head and tail entities caused by **Distance Model** with a relation-specific bilinear form: $f_r(h,t) = \mathbf{h}^T W_r \mathbf{t}$.

Neural Tensor Network (NTN) [14] proposes a general scoring function: $f_r(h,t) = \mathbf{u}_r^T g(\mathbf{h}^T W_r \mathbf{t} + W_{rh}\mathbf{h} + W_{rt}\mathbf{t} + \mathbf{b}_r)$, which combines the **Single Layer Model** and the **Bilinear Model**. This model is more expressive as the second-order correlations are also considered into the nonlinear transformation function, but the computational complexity is rather high.

TransE [12] is the state-of-the-art method so far. Differing from all the other prior arts, this approach embeds relations into the same vector space of entities by regarding the relation r as a translation from h to t, i.e. $\mathbf{h}+\mathbf{r}=\mathbf{t}$. This model works well on the facts with ONE-TO-ONE mapping property, as minimizing the global loss function will impose $\mathbf{h}+\mathbf{r}$ equaling to $\mathbf{t}$. However, the facts with multi-mapping properties, i.e MANY-TO-MANY, MANY-TO-ONE and ONE-TO-MANY, impact the performance of the model. Given a series of facts associated with a ONE-TO-MANY relation r, e.g. $(h,r,t_1),(h,r,t_2),...,(h,r,t_m)$, **TransE** tends to represent the embeddings of entities on MANY-side extremely the same with each other and hardly to be discriminated. Moreover, the learning algorithm of **TransE** only considers the randomly built negative triplets, which may bring in bias for embedding entities and relations.

Therefore, we propose a generic model (**LMNNE**) in the subsequent section to tackle the margin-based knowledge embedding problem. This model can fully take advantage of both positive and negative triplets, and in addition, be flexible enough when dealing with the multi-mapping properties.

III. Proposed Model

Given a triplet (h, r, t), we use the following formula as the scoring function $f_r(h, t)$ to measure its plausibility, i.e.

$$f_r(h,t) = ||h + r - t||, \tag{1}$$

where $||\Box||$ stands for the generic distance metrics (L_1 *norm* or L_2 *norm*) depending on the model selection.

Ideally, we look forward to learning a unique low-dimensional vector representation for each entity and relation, so that all the triplets in the KB will satisfy the equation $\mathbf{h} + \mathbf{r} = \mathbf{t}$. However, this can not be done perfectly because of the subsequent multi-mapping reasons,

- Not all entity pairs involve in only one relation. For example, (*Barack Obama, president_of, U.S.A*) and (*Barack Obama, born_in, U.S.A*) are both correct facts in the KB. However, we do not expect that the embedding of the relation *president_of* is the same as the relation *born_in*, since those relations are not semantically related.
- Besides the multi-relation property above, we may also face the problem of multiple head entities or tail entities when the other two elements are given. For example, there are at least five correct tail entities, i.e. *Comedy film, Adventure film, Science fiction, Fantasy* and *Satire*, given (*WALL-E, has_the_genre, ?*). Likewise, we do not desire that those tail entities share the same embedding.

Therefore, we suggest a 'soft' way of modeling triplets in KBs. Suppose that Δ is the set of facts in the KB. For each triplet (h, r, t) in Δ, we build a set of reconstructed triplets $\Delta_{(h,r,t)}$ by means of replacing the head or the tail with all the other entities in turn. $\Delta_{(h,r,t)}$ can be divided into two sets, i.e. $\Delta^+_{(h,r,t)}$ and $\Delta^-_{(h,r,t)}$. $\Delta^+_{(h,r,t)}$ is the positive set of triplets reconstructed from (h, r, t) as $\Delta^+_{(h,r,t)} \subset \Delta$, and $\Delta^-_{(h,r,t)}$ is the negative because $\Delta^-_{(h,r,t)} \not\subset \Delta$. The intuitive goal of our model is to learn the embeddings of positive tail entities $\mathbf{t}^+$ closer to $\mathbf{h} + \mathbf{r}$ than any other negative embeddings $\mathbf{t}^-$. Therefore, the goal contains two objects, i.e. pulling the positive neighbors near (**NN**) each other while pushing the negatives a large margin (**LM**) away.

Specifically, for a pair of positive triplets, (h, r, t) and (h^+, r, t^+), we *pull* the head or the tail entities together (left panel, Figure 1) by minimizing the loss of variances in distance, i.e.

$$\mathcal{L}_{pull} = \text{Min} \sum_{(h,r,t)\in\Delta} \sum_{(h^+,r,t^+)\in\Delta^+_{(h,r,t)}} (||h - h^+|| + ||t - t^+||). \tag{2}$$

Simultaneously, we *push* the negative head or tail entities away (right panel, Figure 1) via keeping all the negative distances $f_r(h^-, t^-)$ at least one margin γ farther than $f_r(h^+, t^+)$. Therefore, the objective function is,

$$\mathcal{L}_{push} = \text{Min} \sum_{(h,r,t)\in\Delta} \sum_{(h^-,r,t^-)\in\Delta^-_{(h,r,t)}} [\gamma + f_r(h,t) - f_r(h^-,t^-)]_+. \tag{3}$$

$[\Box]_+$ is the hinge loss function equivalent to $max(0, \Box)$, which can spot and update the negative triplets that violate the margin constrains.

Finally, we propose the *Large Margin Nearest Neighbor Embedding* model which uses μ to control the trade-off between the pulling $\mathcal{L}_{pull}$ and pushing $\mathcal{L}_{push}$ operations by setting the total loss $\mathcal{L}$ as follows,

$$\mathcal{L} = \text{Min} \quad \mu\mathcal{L}_{pull} + (1 - \mu)\mathcal{L}_{push}. \tag{4}$$

IV. Learning Algorithm

To efficiently search the optimal solution of **LMNNE**, we use *Stochastic Gradient Descent* (SGD) to update the embeddings of entities and relations in iterative fashion. As shown in Algorithm 1, we firstly initial all the entities and relations following a uniform distribution. Each time we pick a triplet (h, r, t) from Δ, an accompanied triplet (h', r, t') is sampled at the same time by replacing the head or the tail with another entity from the entity set E, i.e. $\Delta'_{(h,r,t)} = \{(h', r, t)|h' \in E\} \cup \{(h, r, t')|t' \in E\}$. Then we choose one of the pair-wise SGD-based updating options depending on which camp ($\Delta^+_{(h,r,t)}$ or $\Delta^-_{(h,r,t)}$) that (h', r, t') belongs to.

Algorithm 1 The Learning Algorithm of **LMNNE**

```
Input:
    Training set $\Delta = \{(h,r,t)\}$, entity set $E$, relation set $R$; dimension of embeddings $d$, margin $\gamma$, learning rate $\alpha$ and $\beta$ for $\mathcal{L}_{pull}$ and $\mathcal{L}_{push}$ respectively, convergence threshold $\epsilon$, maximum epoches $n$ and the trade-off $\mu$.
foreach $\mathbf{r} \in R$ do
    $\mathbf{r}$ := Uniform($\frac{-6}{\sqrt{d}}, \frac{6}{\sqrt{d}}$)
    $\mathbf{r} := \frac{\mathbf{r}}{|\mathbf{r}|}$
end foreach
$i := 0$
while $Rel.loss > \epsilon$ and $i < n$ do
    foreach $\mathbf{e} \in E$ do
        $\mathbf{e}$ := Uniform($\frac{-6}{\sqrt{d}}, \frac{6}{\sqrt{d}}$)
        $\mathbf{e} := \frac{\mathbf{e}}{|\mathbf{e}|}$
    end foreach
    foreach $(h,r,t) \in \Delta$ do
        $(h',r,t')$ := Sampling($\Delta'_{(h,r,t)}$)
        if $(h',r,t') \in \Delta^+_{(h,r,t)}$ then
            Updating: $\bigtriangledown_{(h,r,t,h',t')}\mathcal{L}_{pull}$ with: $\alpha\mu$
        end if
        if $(h',r,t') \in \Delta^-_{(h,r,t)}$ then
            Updating : $\bigtriangledown_{(h,r,t,h',t')}\mathcal{L}_{push}$ with: $\beta(1-\mu)$
        end if
    end foreach
end while
Output:
    All the embeddings of e and r, where $e \in E$ and $r \in R$.
```

V. Quantitative Experiments

Embedding the knowledge into low-dimensional vector spaces makes it much easier for AI-related computing tasks, such as *Link prediction* (predicting t given h and r or h given r and t) and *Triplet classification* (to discriminate whether a triplet (h, r, t) is correct or wrong). Two latest studies [12],

[14] used subsets of WordNet (**WN**) and Freebase (**FB**) data to evaluate their models and reported the performance on the two tasks respectively.

In order to conduct solid experiments, we compare our model (**LMNNE**) with many related studies including state-of-the-art and baseline approaches involving in the two tasks, i.e. *Link prediction* and *Triplet classification*. All the datasets, the source codes and the learnt embeddings for entities and relations can be downloaded from http://1drv.ms/1pUPwzP.

A. *Link prediction*

One of the benefits on knowledge embedding is that we can apply simple vector calculations to many reasoning tasks. For example, *Link prediction* is a valuable task that contributes to completing the knowledge graph. Specifically, It aims at predicting the missing entity or the relation given the other two elements in a mutilated triplet.

With the help of knowledge embeddings, if we would like to tell whether the entity h has the relation r with the entity t, we just need to calculate the distance between $\mathbf{h} + \mathbf{r}$ and $\mathbf{t}$. The closer they are, the more possibility the triplet (h, r, t) exists.

1) Datasets: Bordes et al. [17], [12] released two benchmark datasets[7] which were extracted from WordNet (**WN18**) and Freebase (**FB15K**). Table 1 shows the statistics of these two datasets. The scale of **FB15K** dataset is larger than **WN18** with much more relations but fewer entities.

TABLE I. STATISTICS OF THE DATASETS USED FOR LINK PREDICTION TASK.

DATASET	WN18	FB15K
#(ENTITIES)	40,943	14,951
#(RELATIONS)	18	1,345
#(TRAINING EX.)	141,442	483,142
#(VALIDATING EX.)	5,000	50,000
#(TESTING EX.)	5,000	59,071

2) Evaluation Protocol: For each testing triplet, all the other entities that appear in the training set take turns to replace the head entity. Then we get a bunch of candidate triplets associated with the testing triplet. The dissimilarity of each candidate triplet is firstly computed by the scoring functions, then sorted in ascending order, and finally the rank of the ground truth triplet is recorded. This whole procedure runs in the same way for replacing the tail entity, so that we can gain the mean results. We use two metrics, i.e. *Mean Rank* and *Mean Hit@10* (the proportion of ground truth triplets that rank in Top-10), to measure the performance. However, the results measured by those metrics are relatively inaccurate, as the procedure above tends to generate the false negative triplets. In other words, some of the candidate triplets rank rather higher than the ground truth triplet just because they also appear in the training set. We thus filter out those triplets to report more reasonable results.

3) Experimental Results: We compare our model **LMNNE** with the state-of-the-art **TransE** and other models mentioned in [17] and [18] evaluated on the **WN18** and **FB15K**. We tune the parameters of each previous model[8] based on the validation set, and select the combination of parameters which leads to the best performance. The results are almost the same as [12]. For **LMNNE**, we tried several combinations of parameters: $d = \{20, 50, 100\}$, $\gamma = \{0.1, 1.0, 2.0, 10.0\}$, $\alpha = \{0.01, 0.02, 0.05, 0.1, 0.5, 1.0\}$, $\beta = \{0.01, 0.02, 0.05, 0.1, 0.5, 1.0\}$ and $\mu = \{0.2, 0.4, 0.5, 0.6, 0.8\}$, and finally chose d = 20, γ = 2.0, $\alpha = \beta$ = 0.02 and μ = 0.6 for **WN18** dataset; d = 50, γ = 1.0, $\alpha = \beta$ = 0.02 and μ = 0.6 for **FB15K** dataset. Moreover, we adopted different distance metrics, such as L_1 *norm*, L_2 *norm* and *inner product*, for the scoring function. Experiments show that L_2 *norm* and L_1 *norm* are the best choices to measure the distances in $\mathcal{L}_{pull}$ and $\mathcal{L}_{push}$ for both of the two datasets, respectively. Table 2 demonstrates that **LMNNE** outperforms all the prior arts, including the baseline model **Unstructured** [18], **RESCAL** [19], **SE** [13], **SME (LINEAR)** [18], **SME (BILINEAR)** [18], **LFM** [16] and the state-of-the-art **TransE**[9] [17], [12], measured by *Mean Rank* and *Mean Hit@10.*

Moreover, we divide **FB15K** into different categories (i.e. ONE-TO-ONE, ONE-TO-MANY, MANY-TO-ONE and MANY-TO-MANY) based on the mapping properties of facts. According to **TransE** [12], we set 1.5 as the threshold to discriminate ONE and MANY. For example, given a pair (*h, r*), if the average number of tails appearing in the dataset is upon 1.5, we can categorize the triplets involving in this relation r into the ONE-TO-MANY class. We evaluate the performance of **Filter Hit@10** metric on each category. Table 3 shows that **LMNNE** performs best on most categories. The result proves that the proposed approach can not only maintain the characteristic of modeling the ONE-TO-ONE triplets, but also better handle the facts with multi-mapping properties.

B. *Triplet classification*

Triplet classification is another task proposed by Socher et al. [14] which focuses on searching a relation-specific distance threshold σ_r to tell whether a triplet (h, r, t) is plausible.

1) Datasets: Similar to Bordes et al. [17], [12], Socher et al. [14] also constructed two standard datasets[10], i.e. **WN11** and **FB13**, sampled from WordNet and Freebase. However, both of the datasets contain much fewer relations. Therefore, we build another dataset following the principle proposed by Socher et al. [14] based on **FB15K** which owns much more relations. The head or the tail entity can be randomly replaced with another one to produce a negative triplet, but in order to build much tough validation and testing datasets, the principle emphasizes that the picked entity should once appear at the same position. For example, *(Pablo Picaso,*

[7] The datasets can be downloaded from https://www.hds.utc.fr/everest/doku.php?id=en:transe

[8] All the codes for the related models can be downloaded from https://github.com/glorotxa/SME

[9] To conduct fair comparisons, we re-implemented **TransE** based on the SGD algorithm (not mini-batch SGD) and fed the same random embeddings for initializing both **LMNNE** and **TransE**. That's the reason why our experimental results of **TransE** are slightly different from the original paper [17], [12].

[10] Those datasets can be download from the website http://www.socher.org/index.php

TABLE II. Link prediction results. We compared our proposed LMNNE with the state-of-the-art method (TransE) and other prior arts.

DATASET	WN18				FB15K			
METRIC	*MEAN RANK*		*MEAN HIT@10*		*MEAN RANK*		*MEAN HIT@10*	
	Raw	*Filter*	*Raw*	*Filter*	*Raw*	*Filter*	*Raw*	*Filter*
Unstructured	315	304	35.3%	38.2%	1,074	979	4.5%	6.3%
RESCAL	1,180	1,163	37.2%	52.8%	828	683	28.4%	44.1%
SE	1,011	985	68.5%	80.5%	273	162	28.8%	39.8%
SME (LINEAR)	545	533	65.1%	74.1%	274	154	30.7%	40.8%
SME (BILINEAR)	526	509	54.7%	61.3%	284	158	31.3%	41.3%
LFM	469	456	71.4%	81.6%	283	164	26.0%	33.1%
TransE	294.4	283.2	70.4%	80.2%	243.3	139.9	36.7%	44.3%
LMNNE	**257.3**	**245.4**	**73.7%**	**84.1%**	**221.2**	**107.4**	**38.3%**	**48.2%**

TABLE III. Results of *Filter Hit@10* (in %) on FB15K categorized by different mapping properties of facts (M. stands for MANY).

TASK	*Predicting head*				*Predicting tail*			
REL. Mapping	1-TO-1	1-TO-M.	M.-TO-1	M.-TO-M.	1-TO-1	1-TO-M.	M.-TO-1	M.-TO-M.
Unstructured	34.5%	2.5%	6.1%	6.6%	34.3%	4.2%	1.9%	6.6%
SE	35.6%	62.6%	17.2%	37.5%	34.9%	14.6%	68.3%	41.3%
SME (LINEAR)	35.1%	53.7%	19.0%	40.3%	32.7%	14.9%	61.6%	43.3%
SME (BILINEAR)	30.9%	69.6%	**19.9%**	38.6%	28.2%	13.1%	76.0%	41.8%
TransE	43.7%	65.7%	18.2%	**47.2%**	43.7%	19.7%	66.7%	50.0%
LMNNE	**59.2%**	**77.8%**	17.5%	45.5%	**58.6%**	**20.0%**	**80.9%**	**51.2%**

TABLE IV. Statistics of the datasets used for triplet classification task.

DATASET	WN11	FB13	FB15K
#(ENTITIES)	38,696	75,043	14,951
#(RELATIONS)	11	13	1,345
#(TRAINING EX.)	112,581	316,232	483,142
#(VALIDATING EX.)	5,218	11,816	100,000
#(TESTING EX.)	21,088	47,466	118,142

TABLE V. The accuracy of triplet classification compared with the state-of-the-art method (TransE) and other prior arts.

DATASET	WN11	FB13	FB15K
Distance Model	53.0%	75.2%	-
Hadamard Model	70.0%	63.7%	-
Single Layer Model	69.9%	85.3%	-
Bilinear Model	73.8%	84.3%	-
NTN	70.4%	**87.1%**	66.7%
TransE	77.5%	67.5%	85.8%
LMNNE	**78.6%**	74.8%	**86.8%**

nationality, American) is a potential negative example rather than the obvious irrational *(Pablo Picaso, nationality, Van Gogh)*, given a positive triplet *(Pablo Picaso, nationality, Spainish)*, as *American* and *Spainish* are more common as the tails of *nationality*. Table 4 shows the statistics of the standard datasets that we used for evaluating models on the triplet classification task.

2) Evaluation Protocol: The decision strategy for binary classification is simple: If the dissimilarity of a testing triplet (h, r, t) computed by $f_r(h, t)$ is below the relation-specific threshold σ_r, it is predicted as positive, otherwise negative. The relation-specific threshold σ_r can be searched via maximizing the classification accuracy on the validation triplets which belong to the relation r.

3) Experimental Results: We use the best combination of parameter settings in the *Link prediction* task ($d = 20$, $\gamma = 2.0$, $\alpha = \beta = 0.02$ and $\mu = 0.6$ for **WN11** dataset; $d = 50$, $\gamma = 1.0$, $\alpha = \beta = 0.02$ and $\mu = 0.6$ for both **FB13** and **FB15K** datasets) to generate the entity and relation embeddings, and learn the best classification threshold σ_r for each relation r. Compared with the state-of-the-art, i.e. **TransE** [12], [17] and other prior arts, such as **Distance Model** [13], **Hadamard Model** [20], **Single Layer Model** [14], **Bilinear Model** [15], [16] and **Neural Tensor Network (NTN)**[11] and [14], the proposed **LMNNE** still achieves better performance as shown in Table 5.

VI. Qualitative Analysis

In addition to the quantitative experiments of evaluating the performance on *Link prediction* and *Triplet classification* with several benchmark datasets, we analytically compare the parameter complexity among the approaches that we have mentioned as well. Table 6 lists the theoretical costs on representing triplets (h, r, t) based on the scoring functions of nearly all the classical models. Except for **Unstructured**, **TransE** and **LMNNE**, the other approaches regard the relation r as a transportation matrix serving for the entities h and t. These models need more resources on storing and computing embeddings. **Unstructured** costs least, but does not contain any information on relations. **LMNNE** and **TransE** embed both entities and relations into the low-dimensional vector spaces from different aspects of observations: **LMNNE** regards a relation as the inner connections of word embedding calculations, and **TransE** considers it as a kind of translation from one entity to another. Despite the varies angles of modeling, both of them are relatively efficient models for knowledge representation.

VII. Conclusion and Future Work

Knowledge embedding is an alternative way of representing knowledge besides displaying in triplets, i.e. (h, r, t). Its essence is to learn a distributed representation for each entity and relation, to make the knowledge computable, e.g. $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$.

To achieve higher quality of embeddings, we propose **LMNNE**, a both effective and efficient model on learning

[11] Socher et al. reported higher classification accuracy in [14] with word embeddings. In order to conduct a fair comparison, the accuracy of **NTN** reported in Table 5 is same with the EV (entity vectors) results in Figure 4 of [14].

TABLE VI. THE SCORING FUNCTION AND PARAMETER COMPLEXITY ANALYSIS FOR ALL THE MODELS MENTIONED IN THE EXPERIMENTS. FOR ALL THE MODELS, WE ASSUME THAT THERE ARE A TOTAL OF n_e ENTITIES, n_r RELATIONS (IN MOST CASES, $n_e \gg n_r$.), AND EACH ENTITY IS EMBEDDED INTO A d-DIMENSIONAL VECTOR SPACE, I.E $\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$. WE ALSO SUPPOSE THAT THERE ARE s SLICES IN A TENSOR FOR THE NEURAL-NETWORK RELATED MODELS, I.E *Single Layer Model* AND *Neural Tensor Network*.

Model	**Scoring Function**	**Parameter Complexity**
Unstructured	$\|\|\mathbf{h} - \mathbf{t}\|\|$	$n_e d$
Distance Model (SE)	$\|\|W_{rh}\mathbf{h} - W_{rt}\mathbf{t}\|\|$; $(W_{rh}, W_{rt}) \in \mathbb{R}^{d \times d}$	$n_e d + 2n_r d^2$
Single Layer Model	$\mathbf{u}_r^T \tanh(W_{rh}\mathbf{h} + W_{rt}\mathbf{t} + \mathbf{b}_r)$; $(W_{rh}, W_{rt}) \in \mathbb{R}^{s \times d}$, $(\mathbf{u_r}, \mathbf{b}_r) \in \mathbb{R}^s$	$n_e d + 2n_r(sd + s)$
Bilinear Model	$\mathbf{h}^T W_r \mathbf{t}$; $W_r \in \mathbb{R}^{d \times d}$	$n_e d + n_r d^2$
Neural Tensor Network (NTN)	$\mathbf{u}_r^T \tanh(\mathbf{h}^T W_r \mathbf{t} + W_{rh}\mathbf{h} + W_{rt}\mathbf{t} + \mathbf{b}_r)$; $W_r \in \mathbb{R}^{d \times d \times s}$, $(W_{rh}, W_{rt}) \in \mathbb{R}^{s \times d}$, $(\mathbf{u_r}, \mathbf{b}_r) \in \mathbb{R}^s$	$n_e d + n_r(sd^2 + 2sd + 2s)$
TransE and **LMNNE**	$\|\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|\|$; $\mathbf{r} \in \mathbb{R}^d$	$n_e d + n_r d$

a low-dimensional vector representation for each entity and relation in Knowledge Bases. Some canonical tasks, such as *Link prediction* and *Triplet classification*, which were ever based on hand-made logic rules, can be truly facilitated by means of the simple vector calculation. The results of extensive experiments on several benchmark datasets and complexity analysis show that our model can achieve higher performance without sacrificing efficiency.

In the future, we look forward to paralleling the algorithm which can encode a whole KB with billion of facts, such as Freebase and YAGO. Another direction is that we can apply this new way of *Knowledge Representation* on reinforcing some other related studies, such as *Relation Extraction* from free texts and *Open Question Answering*.

VIII. ACKNOWLEDGMENTS

This work is supported by National Program on Key Basic Research Project (973 Program) under Grant 2013CB329304, National Science Foundation of China (NSFC) under Grant No.61373075. The first author conducted this research while he was a joint-supervision Ph.D. student in New York University. This paper is dedicated to all the members of the Proteus Project[12], and thanks so much to your help.

REFERENCES

[1] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[2] D. Lenat, "Cyc: A large-scale investment in knowledge infrastructure," *Communications of the ACM*, vol. 38, no. 11, pp. 33–38, November 1995.

[3] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A Core of Semantic Knowledge," in *16th international World Wide Web conference (WWW 2007)*. New York, NY, USA: ACM Press, 2007.

[4] K. Bollacker, R. Cook, and P. Tufts, "Freebase: A shared database of structured general human knowledge," in *AAAI*, vol. 7, 2007, pp. 1962–1963. [Online]. Available: http://www.aaai.org/Papers/AAAI/2007/AAAI07-355.pdf

[5] D. C. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification." in *CVPR*. IEEE, 2012, pp. 3642–3649. [Online]. Available: http://dblp.uni-trier.de/db/conf/cvpr/cvpr2012.html#CiresanMS12

[6] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *arXiv preprint arXiv:1405.4053*, 2014.

[7] H. Schwenk and J.-L. Gauvain, "Connectionist language modeling for large vocabulary continuous speech recognition." in *ICASSP*. IEEE, 2002, pp. 765–768. [Online]. Available: http://dblp.uni-trier.de/db/conf/icassp/icassp2002.html#SchwenkG02

[8] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model." *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003. [Online]. Available: http://dblp.uni-trier.de/db/journals/jmlr/jmlr3.html#BengioDVJ03

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[10] T. Mikolov, W. tau Yih, and G. Zweig, "Linguistic regularities in continuous space word representations." in *HLT-NAACL*. The Association for Computational Linguistics, 2013, pp. 746–751. [Online]. Available: http://dblp.uni-trier.de/db/conf/naacl/naacl2013.html#MikolovYZ13

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., 2013, pp. 3111–3119.

[12] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Advances in Neural Information Processing Systems*, 2013, pp. 2787–2795.

[13] A. Bordes, J. Weston, R. Collobert, Y. Bengio *et al.*, "Learning structured embeddings of knowledge bases." in *AAAI*, 2011. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/viewPDFInterstitial/3659/3898

[14] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *Advances in Neural Information Processing Systems*, 2013, pp. 926–934.

[15] I. Sutskever, R. Salakhutdinov, and J. B. Tenenbaum, "Modelling relational data using bayesian clustered tensor factorization." in *NIPS*, 2009, pp. 1821–1828.

[16] R. Jenatton, N. Le Roux, A. Bordes, G. Obozinski *et al.*, "A latent factor model for highly multi-relational data." in *NIPS*, 2012, pp. 3176–3184.

[17] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Irreflexive and hierarchical relations as translations," *arXiv preprint arXiv:1304.7158*, 2013. [Online]. Available: http://arxiv.org/abs/1304.7158

[18] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "A semantic matching energy function for learning with multi-relational data," *Machine Learning*, vol. 94, no. 2, pp. 233–259, 2014. [Online]. Available: http://link.springer.com/article/10.1007/s10994-013-5363-6

[19] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 809–816.

[20] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "Joint learning of words and meaning representations for open-text semantic parsing," in *International Conference on Artificial Intelligence and Statistics*, 2012, pp. 127–135.

[12] http://nlp.cs.nyu.edu/index.shtml